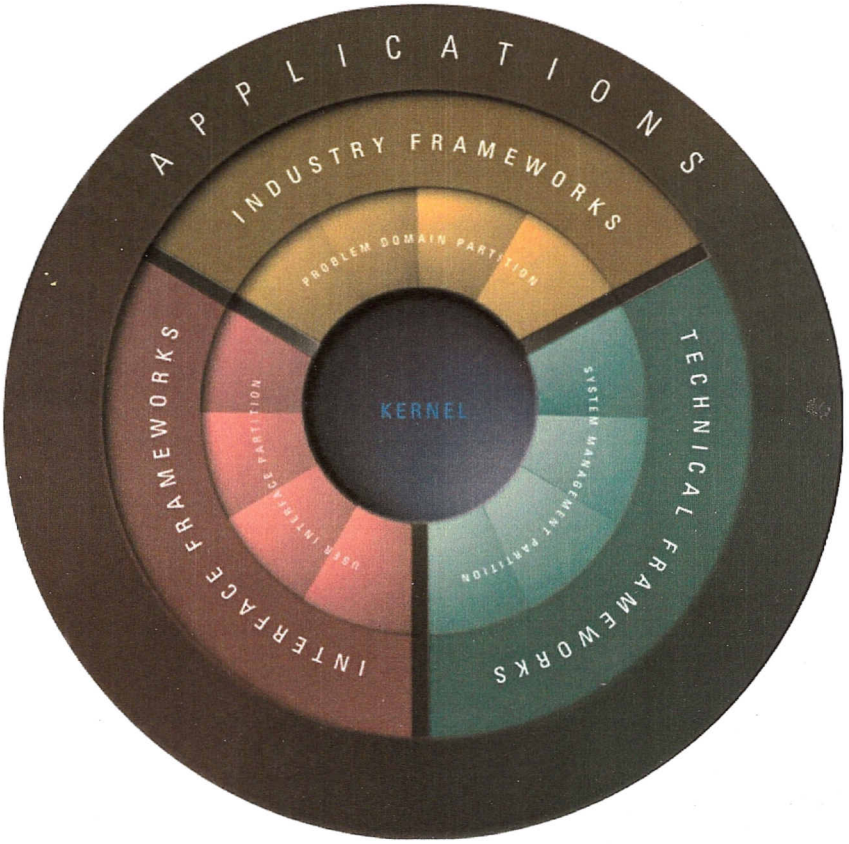
AdAPT

Advanced Application Partitioning Toolkit

Application Developer's Technical Reference



AUSTIN SOFTWARE FOUNDRY

AUSTIN SOFTWARE FOUNDRY

AdAPT

Application Developer's Technical Reference Guide

Version 1.2, August 1997

The software described in this manual is provided by Austin Software Foundry, Inc. (ASF) under a License Agreement. The software may be used only in accordance with the terms of the License Agreement. Information in this manual may change without notice and does not represent a commitment on the part of ASF.

Copyright 1996,1997, Austin Software Foundry

Austin Software Foundry (ASF) claims copyright of this documentation as an unpublished work, revisions of which were first licensed on the date indicated on the date in the foregoing notice. Claim of copyright does not imply waiver of ASF's other rights.

Printed: August, 1997

Austin Software Foundry
Capital of Texas Highway, North
Building 8, Suite 250
Austin, Texas 78746
(512) 329 6697
www.foundry.com

TABLE OF CONTENTS

Introduction I
Part One: Base Classes 1
ASFAdapter 4
ASFAdapterManager 6
GetAdapter() 8
ASFAttribute 9
GetAttributeName() 11
SetAttributeName() 13
GetAttributeValue() 15
SetAttributeValue() 16
ASFFactory 17
CreateObject() 19
ASFMessageArguments 20
Clear() 22
ASFMessageStringArguments 23
ASFMessageStringArguments() 25
GetValue() 26
GetName() 27
GetFirstArgument() 28
GetLastArgument() 29
GetNextArgument() 30
GetPreviousArgument() 31
GetValueWithName() 32
SetName() 33
SetValue() 34
Destructor() 35
ASFObjectFactory 36
CreateObject() 38
ASFService 39
PerfomService() 41
ASFStringAttribute 42
GetAttributeValue() 44
SetAttributeValue() 46
ASFSystemFunctions 48
StringToBoolean() 49
StringToArray() 51
IsValidObject() 52
The Problem Domain Partition 53
ASFBusinessContext 54
RequestDataWindow() 56
ProcessTaskRequest() 58
ParseTaskRequest() 60
ShareDataStore() 62

pronf_register_business_object() 64
pronf_unregister_business_object() 66
Constructor() 68
Destructor() 69
ASFBusinessEntity 70
BeginQueryMode() 73
RequestDataStore() 75
RegisterDataWindow() 76
RequestDataStore() 78
DeleteCurrentRow() 79
DeleteSelectedRows() 81
DuplicateRow() 83
EndQueryMode() 85
GetQueryMode() 87
InsertRowAfter() 89
InsertRowBefore() 91
MoveToFirstRow() 93
MoveToLastRow() 95
MoveToNextRow() 97
MoveToPreviousRow() 99
RetrieveData() 101
ToggleSelectedRow() 102
ShareDataStore() 103
pronf_register_adapter() 105
pronf_unregister_adapter() 106
Constructor() 107
Destructor() 108
ASFBusinessRule 109
ASFBusinessAdapter 110
ASFBusinessEntityDWC 111
BeginQueryMode() 113
DeleteCurrentRow() 114
DeleteSelectedRows() 115
DuplicateRow() 116
EndQueryMode() 117
GetQueryMode() 118
InsertRowAfter() 119
InsertRowAfter() 120
MoveToFirstRow() 121
MoveToLastRow() 122
MoveToNextRow() 123
MoveToPreviousRow() 124
RetrieveData() 125
ToggleSelectedRow() 126
Constructor() 127
..... 128

The System Management Partition 129
ASFApplicationManager 130
UserRequest() 132
ProcessTaskRequest() 133
ASFComponentManagerFacade 134
GetCount() 136
Register() 137
Register() 139
Unregister() 142
RequestComponent() 143
MoveToFirst() 144
MoveToNext() 146
GetComponent() 148
Constructor() 150
Destructor() 151
ASFDataSourceManager 152
GetDataSourceAdapter() 154
RegisterAdapter() 155
Constructor() 157
Destructor() 158
ASFDataStoreManager 159
RequestDataStore() 161
ASFNamedComponentManagerFacade 162
GetCount() 164
Register() 165
Register() 168
Unregister() 171
RequestComponent() 172
MoveToFirst() 173
MoveToNext() 175
GetComponent() 177
Constructor() 179
Destructor() 180
ASFNamedServiceManagerFacade 181
GetCount() 183
Register() 184
Register() 186
RequestService() 189
Unregister() 191
Constructor() 192
Destructor() 193
ASFServiceManagerFacade 194
GetCount() 196
Register() 197
Register() 199
RequestService() 202

Unregister() ... 204
Constructor() ... 205
Destructor() ... 206
ASFTransactionManager ... 207
ConnectToDB() ... 209
DisconnectFromDB() ... 210
RequestTransaction() ... 211
ASFResourceAdapter ... 212
OpenResource() ... 214
SetSection() ... 215
SetTransactionObject() ... 216
DeleteEntry() ... 217
RretrieveValue() ... 218
UpdateValue() ... 219
ASFResourceDatabaseAdapter ... 220
SetSection() ... 222
prinf_fetch_db_data() ... 223
DeleteEntry() ... 224
RetrieveValue() ... 225
UpdateValue() ... 226
OpenResource() ... 227
SetTransactionObject() ... 228
ASFResourceFacade ... 229
OpenResource() ... 231
SetSection() ... 233
SetTransactionObject() ... 234
DeleteEntry() ... 235
RetrieveValue() ... 236
UpdateValue() ... 237
SetAdapter() ... 238
Destructor() ... 239
ASFResourceIniAdapter ... 240
OpenResource() ... 242
DeleteEntry() ... 243
RetrieveValue() ... 244
UpdateValue() ... 245
SetSection() ... 246
ASFResourceRegistryAdapter ... 247
OpenResource() ... 249
SetSection() ... 250
DeleteEntry() ... 251
RetrieveValue() ... 252
UpdateValue() ... 253
ASFController ... 254
RegisterDataWindow() ... 256
nf_parse_task_request() ... 258

ProcessTaskRequest() ... 260
ParseTaskRequest() ... 262
The User Interface Partition ... 263
ASFContainerWindow ... 264
ASFDialogWindow ... 265
ASFViewWindow ... 266
ASFWindow ... 268
GetWindowName() ... 270
SetWindowName() ... 271
ParseTaskRequest() ... 272
ProcessTaskRequest() ... 273
ProcessUserRequest() ... 274
Open() ... 275
ASFContainerMenu ... 276
ASFMenu ... 277
MenuItemClicked() ... 278
ASFPopupMenu ... 279
ASFViewMenu ... 280
ASFCustomUserObject ... 281
ASFDataWindowControl ... 282
MoveToFirstRow() ... 284
MoveToNextRow() ... 285
MoveToPreviousRow() ... 286
MoveToLastRow() ... 287
DeleteCurrentRow() ... 288
DeleteSelectedRows() ... 289
RetieveData() ... 290
BeginQueryMode() ... 291
EndQueryMode() ... 292
GetQueryMode() ... 293
DuplicateRow() ... 294
UpdateData() ... 295
InsertRowAfter() ... 296
InsertRowBefore() ... 297
SetDataWindowControl() ... 298
GetDataWindowControlName() ... 299
ASFDataWindowControlManager ... 300
RequestDataWindow() ... 301
Part Two: Components ... 303
The Log Component ... 305
ASFLogAdapter ... 306
CloseLog() ... 308
EraseLog() ... 309
LogText() ... 310
OpenLog() ... 311

RetrieveAttributes() ... 312
SetAttribute() ... 313
SetIncludeDate() ... 314
SetIncludeTime() ... 315
SetTitle() ... 316
Constructor() ... 317
Destructor() ... 318
ASFLogBaseFacade ... 319
CloseLog() ... 321
EraseLog() ... 322
OpenLog() ... 323
RetrieveAttributes() ... 324
SetIncludeDate() ... 325
SetIncludeTime() ... 326
SetAttribute() ... 327
SetTitle() ... 328
pronf_log_text() ... 329
SetAdapter() ... 330
Constructor() ... 331
Destructor() ... 332
ASFLogFileAdapter ... 333
CloseLog() ... 335
EraseLog() ... 336
LogText() ... 337
OpenLog() ... 338
RetrieveAttributes() ... 339
SetAttribute() ... 340
ASFLogWindow ... 341
wf_clear_window() ... 342
wf_log_text() ... 343
ASFLogWindowAdapter ... 344
CloseLog() ... 346
EraseLog() ... 347
LogText() ... 348
OpenLog() ... 349
RetrieveAttributes() ... 350
SetAttribute() ... 351
ASFLogWindowBase ... 352
wf_clear_window() ... 353
wf_log_text() ... 354
The Message-Parser Component ... 355
ASFMessageParserAdapter ... 356
ParseMessage() ... 358
ASFMessageParserCGIAdapter ... 359
ParseMessage() ... 361
Constructor() ... 362

Destructor() ... 363
ASFMessageParserFacade ... 364
ParseMessage() ... 366
SetAdapter() ... 367
Constructor() ... 368
Destructor() ... 369
ASFMessageParserSimpleAdapter ... 370
ParseMessage() ... 372
Constructor() ... 373
Destructor() ... 374
ASFParser ... 375
ParseMessage() ... 377
pronf_convert_escape_chars() ... 378
ASFParserCGI ... 379
ParseMessage() ... 382
pronv_convert_escape_chars() ... 384
ipronv_hex_string_to_char() ... 385
ASFParserSimple ... 386
ParseMessage() ... 388
The Ad hoc Query Component ... 391
ASFAdHocQueryComponentsFactory ... 392
Initialize() ... 394
Constructor() ... 395
ASFAdHocQueryFacade ... 396
GenerateQueryStatement() ... 398
GenerateQueryStatement() ... 399
GenerateQueryStatement() ... 400
GenerateQueryStatement() ... 401
UnregisterQueryLanaguage() ... 402
RegisterQuery() ... 403
UnregisterQuery() ... 404
CreateQueryObject() ... 405
DestroyQueryObject() ... 406
RequestQuery() ... 407
GetQueryLanguageTypes() ... 408
RegisterQueryLanguage() ... 409
SetDefaultLanguage() ... 410
GetDefaultLanguage() ... 411
GetRegisteredQueryNames() ... 412
Constructor() ... 413
Destructor() ... 414
ASFQuery ... 415
ClearQueryClause() ... 418
ClearAllQueryClauses() ... 419
GetQueryStatementText() ... 420
SetQueryStatementText() ... 421

GetQueryElementTypes() ... 422
AddQueryElement() ... 423
MoveToFirstQueryElement() ... 424
MoveToNextQueryElement() ... 425
GetQueryElement() ... 426
ApendQueryStatementText() ... 427
GetQueryElementType() ... 428
GetQueryElementValue() ... 429
GetQueryName() ... 430
SetQueryName() ... 431
Constructor() ... 432
Destructor() ... 433
ASFQueryElement ... 434
GetQueryElementType() ... 436
SetQueryElementType() ... 437
GetQueryElementValue() ... 438
SetQueryElementValue() ... 439
Constructor() ... 440
ASFQueryElementTypes ... 441
ASFQueryGeneratorAdapter ... 444
GenerateQueryStatement() ... 446
Initialize() ... 447
ASFQueryGeneratorAdapterFactory ... 448
CreateObject() ... 450
ASFQueryGeneratorSQLOQL ... 451
Initialize() ... 454
GenerateQueryStatement() ... 455
pronf_generate_clause() ... 456
Destructor() ... 457
ASFQueryGeneratorSQLOQLAdapter ... 458
GenerateQueryStatement() ... 460
Initialize() ... 461
Destructor() ... 462
ASFQueryLanguageTypes ... 463
ASFQueryStatement ... 465
SetLanguage() ... 467
GetLanguage() ... 468
GetStatement() ... 469
AppendClauseToStatement() ... 470
AppendStatementDelimiter() ... 471
Clear() ... 472
ClearLanguage() ... 473
ClearStatement() ... 474
Constructor() ... 475
Part Three: Kits ... 477
The Publish & Subscribe Kit ... 479

ASFPublisher 480
AttachSubscriber() 482
DetachSubscriber() 483
NotifySubscribers() 484
DetachAllSubscribers() 485
ASFPublisherDistributed 486
AttachSubscriber() 488
DetachSubscriber() 489
NotifySubscribers() 490
DetachAllSubscribers() 491
ASFSubscriber 492
UpdateSubscriber() 494
GetRegistrationID() 495
SetRegistrationID() 496
The Multiple Document Interface Kit 497
ASFMDIFrameMenu 498
RegisterFrame() 499
ASFMDIFrameWindow 500
ASFMDISheetMenu 501
ASFMDISheetWindow 502

INTRODUCTION

About This Manual

This manual provides information that is crucial for learning about the concepts and tools for the Advanced Application Partitioning Toolkit (AdAPT). This manual contains a conceptual foundation for application developers on the AdAPT product and its library of object-oriented classes. Much of the information presented here will give the developer insight into using the AdAPT Application Developer's Technical Reference Guide. It is therefore suggested that this manual and the Technical Reference be used in tandem with each other.

How the Manual Is Organized

Each part of this manual is dedicated to describing the classes of ASF's Application Partitioning Model. The sections include:

Part One: Base Classes

Part Two: Components

Part Three: Kits

How the Chapters Are Organized

Each part describes classes. Under the class name, several subheadings describe the characteristics of the class. The subheadings include:

Ancestor Hierarchy

This section shows the ancestry of the class from the PowerBuilder ancestor to the class itself.

Declared In

This section shows the PowerBuilder Library (PBL) in which the class is defined.

Class Description

This section includes a description of this class' general purpose, when and where it should be used.

Instance Variables

This section describes the PRIVATE, PROTECTED, and PUBLIC instance variables in the class.

Class Invariant

This section defines what the valid internal and external states are for this class.

External

This subsection describes any application states external to the class that must be true in order for this class to function properly. For example, this subsection would describe any global variables that are required for the proper functioning of this class.

Internal

This subsection describes valid states for this class. This subsection describes the range of valid values or states of the instance variables (class attributes). In other words, if a function went through and tested the values of all of the attributes of an instance of this class at some random point during its execution, what range would each of their values have to be in order for this class to be in a valid state?

Functions

Functions belonging to particular class are grouped together under the Functions subheading.

The functions are in alphabetical order and contain the following subheadings:

Description

This section describes what the function does and what it is used for.

Parameters

This section gives the parameter name, type, and describes the purpose of each parameter of the function.

Returns

This section describes the type of the return value returned by the function and the possible values that can be returned.

Precondition

This section specifies the requirements that the caller must fulfill so that the function executes without producing an exception.

Postcondition

This section specifies what state the class the class is left in once the function has completed executing providing that the precondition was satisfied.

Exception

This section specifies what the function will do if it encounters an exception, an error, an invalid state.

Example

This section gives an example of how to use the function.

See Also

This section specifies where you can find similar classes or functions.

Special Note

This section contains additional helpful information.

Suggested Use

This section includes suggested uses for the class of function.

Method Categories

This section contains a list of all functions conatined in the class.

PART ONE: BASE CLASSES

The Kernel

ASFAdapter (abstract)

Ancestor Hierarchy

nonvisualobject

- **ASFObject**
  - **ASFKernel**
    - **ASFAdapter**

Declared In

ap1kernl.pbl

Class Description

The ASFAdapter class provides the base ancestor for all adapter classes. This is an abastract class and should be inherited from.

Attributes

None.

Relations

None.

Class Invariant

External

None.

Internal

None.

Suggested Uses

Adapt to external systems
Adapt to legacy systems
Adapt to specific class variants

See Also

Log Component
Message Parser Component

Ad hoc Query Component
Data Source Adapters

ASFAdapterManager

Ancestor Hierarchy

nonvisualobject

- **ASFObject**
  - **ASFFacade**
    - **ASFSystemFacade**
      - **ASFManagerFacade**
        - **ASFComponentManagerFacade**
          - **ASFAdapterManager**

Declared In

ap1kernl.pbl

Class Description

AdAPT Adapter Manager class used to manage a set of adapters. The adapters are not named. When an adapter is registered, the registration ID must be saved by the object requesting the registration.

Attributes

None.

Relations

None.

Class Invariant

External

None.

Internal

None.

Suggested Uses

Use where it's necessary to manage several unnamed adapters at once.

See Also

ASFAdapter
ASFComponentManagerFacade

Method Categories

Misc

GetAdapter()

Functions

GetAdapter()

PUBLIC GetAdapter (as_registration_id : STRING, aany_requestor : ANY) : ASFADAPTER

Description

Gets a reference to the adapter with the name as_registration_id.

Parameters

as_registration_id	STRING - Registration ID of the requested adapter.
aany_requestor	ANY - Reference to the requesting object.

Returns

ASFADAPTER	Reference to the adapter to be registered. Null is returned if the adapter cannot be found.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

Example

```
/* instance variables */
ASFAdapterManager ipronv_adapter_mgr
. . .
/* assume as_registration_id is an argument */
ASFAdapter   lnv_adapter
ANY          lany_this

lany_this = THIS

lnv_adapter = ipronv_adapter_mgr.TRIGGER STATIC FUNCTION &
   GetAdapter( as_registration_id, lany_this )
```

ASFAttribute (abstract)

Ancestor Hierarchy

nonvisualobject

- **ASFObject**
  - **ASFKernel**
    - **ASFUtility**
      - **ASFAttribute**

Declared In

ap1kernl.pbl

Class Description

The ASFAttribute class is the base class for specific attribute classes. It is an abstract class and is intended to be inherited from.

Attributes

Protected

ipros_name STRING - The name of the attribute.

Relations

None.

Class Invariant

External

None.

Internal

None.

Suggested Uses

Name-value pairs needed

See Also

ASFStringAttribute
ASFMessageArguments

Method Categories

Get

GetAttributeValue()
GetAttributename()

Set

SetAttributeName()
SetAttributeValue()

Functions

GetAttributeName()

PUBLIC GetAttributename () : STRING

Description

This method gets the name of the attribute.

Parameters

None.

Returns

STRING	Attribute Name. In case of error, the empty string "" is returned.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

Example

```
ASFStringAttribute   lnv_attribute
STRING               ls_attribute value
STRING               ls_attribute_name
BOOLEAN              lb_return

lnv_attribute = CREATE ASFStringAttribute

lb_return = lnv_attribute.TRIGGER STATIC FUNCTION &
   SetAttributeName( "The Attribute Name" )

lb_return = lnv_attribute.TRIGGER STATIC FUNCTION &
   SetAttributeValue( "Some attribute value" )
. . .
ls_attribute_name = lnv_attribute.TRIGGER STATIC FUNCTION &
   GetAttributeName()

ls_attribute_value = lnv_attribute.TRIGGER STATIC FUNCTION &
```

```
GetAttributeValue()
```

See Also

SetAttributeName()

SetAttributeName()

PUBLIC SetAttributeName (as_name : STRING) : BOOLEAN

Description

This method sets the attribute name.

Parameters

as_name	STRING - The name to which to set the attribute name.

Returns

BOOLEAN	TRUE if successful, FALSE otherwise.

Precondition

See class invariant.

Postcondition

The value of ipros_name is set to as_name.

Exception

None.

Example

```
ASFStringAttribute   lnv_attribute
STRING               ls_attribute value
STRING               ls_attribute_name
BOOLEAN              lb_return

lnv_attribute = CREATE ASFStringAttribute

lb_return = lnv_attribute.TRIGGER STATIC FUNCTION &
   SetAttributeName( "The Attribute Name" )

lb_return = lnv_attribute.TRIGGER STATIC FUNCTION &
   SetAttributeValue( "Some attribute value" )
. . .
ls_attribute_name = lnv_attribute.TRIGGER STATIC FUNCTION &
   GetAttributeName()

ls_attribute_value = lnv_attribute.TRIGGER STATIC FUNCTION &
   GetAttributeValue()
```

See Also

GetAttributeName()

GetAttributeValue() (virtual)

PUBLIC GetAttributeValue () : STRING

Description

This method is intended to get the attribute value.

Parameters

None.

Returns

STRING	This method will always return an empty string "".

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

See Also

SetAttributeValue()

SetAttributeValue() (virtual)

PUBLIC SetAttributeValue (as_value : STRING) : BOOLEAN

Description

This method is intended to set the attribute value.

Parameters

as_value	STRING - Attribute value.

Returns

BOOLEAN	This method will always return FALSE

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

See Also

SetAttributeName()

ASFFactory (abstract)

Ancestor Hierarchy

nonvisualobject

- **ASFObject**
  - **ASFKernel**
    - **ASFCreation**
      - **ASFFactory**

Declared In

ap1kernl.pbl

Class Description

The ASFFactory class is the base class for all AdAPT object factories and builder classes. This class is an abstract class and not meant to be implemented directly.

Attributes

None.

Relations

None.

Class Invariant

External

None.

Internal

None.

Suggested Uses

Base class for object factories and builders.

See Also

ASFObjectFactory

Method Categories

Creation

CreateObject()

Functions

CreateObject() (virtual)

PUBLIC CreateObject (as_object_type : STRING) : ANY

Description

Create an object using the argument as_object_type and return a reference to that object.

Parameters

as_object_type	STRING - Object type. This string may be a class name or some other object identifier. It is up to the object factory or builder to properly interpret the string.

Returns

ANY	This method will always return NULL.

Precondition

See class invariants.

Postcondition

See class invariants.

Exception

None.

ASFMessageArguments

Ancestor Hierarchy

nonvisualobject

- **ASFObject**
  - **ASFKernel**
    - **ASFUtility**
      - **ASFMessageArguments**

Declared In

ap1kernl.pbl

Class Description

Attributes

Protected

iproi_current_arg	INTEGER - Index to the current argument (attribute name-value pair).

Relations

Protected

ipronv_attributes	ASFAttribute - Array of attributes (name-value pairs).
ipronv_attributes	ASFAttribute -

Class Invariant

External

None.

Internal

None.

Suggested Uses

Message Parsers

See Also

ASFMessageStringArguments
ASFAttribute
ASFMessageParserFacade

Method Categories

Misc.

Clear()

Functions

Clear()

PUBLIC Clear () : BOOLEAN

Description

The method clears the argument array, ipronv_arguments[].

Parameters

None.

Returns

BOOLEAN	TRUE if successful, FALSE otherwise.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

Example

```
/* Instance Variables */
/* ASFMessageStringArguments is a
   descendent of ASFMessageArguments */
ASFMessageStringArguments ipronv_args
. . .
BOOLEAN  lb_return

lb_return = ipronv_args.TRIGGER STATIC FUNCTION &
   Clear()
```

See Also

ASFMessageStringArguments

ASFMessageStringArguments

Ancestor Hierarchy

nonvisualobject

- **ASFObject**
  - **ASFKernel**
    - **ASFUtility**
      - **ASFMessageArguments**
        - **ASFMessageStringArguments**

Declared In

ap1kernl.pbl

Class Description

The ASFMessageStringArguments class provides access to message string arguments. Arguments are stored in the order in which they were placed in the ASFMessageStringArguments type object. Arguments are stored as attributes (name-value pairs). Both the names and values are strings.

Attributes

None.

Relations

None.

Class Invariant

External

None.

Internal

None.

Suggested Uses

Message Parsing

See Also

ASFMessageArguments
ASFAttribute
ASFStringAttribute

Method Categories

Events

Destructor()

Get

GetFirstArgument()
GetLastArgument()
GetName()
GetNextArgument()
GetPreviousArgument()
GetValue()
GetValueWithName()

Set

SetName()
SetValue()

Store

ASFMessageStringArguments()

Functions

ASFMessageStringArguments()

PUBLIC ASFMessageStringArguments (as_name : STRING, as_value : STRING) : BOOLEAN

Description

Store the name-value pair, as_name and as_value, in the arguments list.

Parameters

as_name	STRING - The argument name.
as_value	STRING - The argument value.

Returns

BOOLEAN	TRUE if successful, FALSE otherwise.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

GetValue()

PUBLIC GetValue (ai_index : INTEGER) : STRING

Description

Get the value of the attribute at the index as_index.

Parameters

ai_index	INTEGER - The index of the argument for which the value is to be returned.

Returns

STRING	The value of the indexed argument

Precondition

ai_index > = 1

ai_index <= UpperBound(ipronv_arguments)

Postcondition

See class invariant.

Exception

None.

See Also

SetValue()

GetName()

SetName()

GetName()

PUBLIC GetName (ai_index : INTEGER) : STRING

Description

Get the name of the attribute at the index as_index.

Parameters

ai_index	INTEGER - The index of the argument name to be returned.

Returns

STRING	The name of the attribute.

Precondition

ai_index >= 1

ai_index <= UpperBound(ipronv_arguments)

Postcondition

See class invariant.

Exception

None.

GetFirstArgument()

PUBLIC GetFirstArgument (as_name : STRING, as_value : STRING) : BOOLEAN

Description

Get the value and name of the first argument in the array ipronv_arguments[]. The value and name are returned in the string arguments as_name and as_value, respectively.

This method also sets the current argument index to 1 (one).

Parameters

as_name	STRING - The argument name.
as_value	STRING - The argument value.

Returns

BOOLEAN	TRUE if successful, FALSE if there are no arguments or either of the input strings is NULL.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

GetLastArgument()

PUBLIC GetLastArgument (as_name : STRING, as_value : STRING) : BOOLEAN

Description

Get the value and name of the last argument in the array ipronv_arguments[]. The value and name are returned in the string arguments as_name and as_value, respectively.

This method also sets the current argument index to the upper bound of the argument array.

Parameters

as_name	STRING - The argument name.
as_value	STRING - The argument value.

Returns

BOOLEAN	TRUE if successful, FALSE if there are no arguments or either of the input strings is NULL.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

GetNextArgument()

PUBLIC GetNextArgument (as_name : STRING, as_value : STRING) : BOOLEAN

Description

Get the value and name of the next argument in the array ipronv_arguments[]. The value and name are returned in the string arguments as_name and as_value, respectively.

This method also increments the current index by one.

If the current argument index is already set to the upper bound of the argument array, then FALSE is returned and the current argument is not changed.

Parameters

as_name	STRING - The argument name.
as_value	STRING - The argument value.

Returns

BOOLEAN	TRUE if successful, FALSE if there are no arguments, or either of the input strings is NULL, or the current argument index is at the upper bound of the argument array

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

GetPreviousArgument()

PUBLIC GetPreviousArgument (as_name : STRING, as_value : STRING) : BOOLEAN

Description

Get the value and name of the next argument in the array ipronv_arguments[]. The value and name are returned in the string arguments as_name and as_value, respectively.

This method also decrements the current index by one.

If the current argument index is already set to the lower bound of the argument array, then FALSE is returned and the current argument is not changed.

Parameters

as_name	STRING - The argument name.
as_value	STRING - The argument value.

Returns

BOOLEAN	TRUE if successful, FALSE if there are no arguments, or either of the input strings is NULL, or the current argument index is at the lower bound of the argument array

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

GetValueWithName()

PUBLIC GetValueWithName (as_name : STRING, as_value : STRING) : BOOLEAN

Description

Get the argument value associated with the name as_name. The value is returned in the method argument as_value.

If found, the current argument index is the argument with the name as_name. If not found, then the current argument index is not changed.

Parameters

as_name	STRING - The argument name.
as_value	STRING - The argument value.

Returns

BOOLEAN	TRUE if successful, FALSE if the argument is not found or the input strings are NULL.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

SetName()

PUBLIC SetName (ai_index : INTEGER, as_name : STRING)

Description

This method sets the name of the attribute at the index ai_index.

Parameters

ai_index	INTEGER - The argument index.
as_name	STRING - The argument name.

Returns

None.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

SetValue()

PUBLIC SetValue (ai_index : INTEGER, as_name : STRING) : BOOLEAN

Description

Set the value of the attribute at the index ai_index.

Parameters

ai_index	INTEGER - The argument index.
as_name	STRING - The argument name.

Returns

BOOLEAN	TRUE if successful, FALSE otherwise.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

Events

Destructor()

PUBLIC Destructor ()

Description

Destroy all attribute object.

Parameters

None.

Returns

None.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

ASFObjectFactory

Ancestor Hierarchy

nonvisualobject

- **ASFObject**
  - **ASFKernel**
    - **ASFCreation**
      - **ASFFactory**
        - **ASFObjectFactory**

Declared In

ap1kernl.pbl

Class Description

The ASFObjectFactory is a generic class for creating any type of user object. This class is used when the class of the object being created is not known. If the class type is known, a descendent of this class should be made and the CreateObject() method overridden.

Attributes

None.

Relations

None.

Class Invariant

External

None.

Internal

None.

Suggested Uses

Use when the class of the object being created is not known.

See Also

ASFFactory

Method Categories

Creation

CreateObject()

Functions

CreateObject()

PUBLIC CreateObject (as_object_type : STRING) : ANY

Description

This method creates an object of type as_object_type , provided that the object type is defined within the application. If the object type is not defined, then a run-time error will occur.

Parameters

as_object_type	STRING - The type of object to create. The object type must be defined within the application.

Returns

ANY	Reference to an object of type as_object_type.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

If the object type is not defined within the application, a run-time error will occur.

Example

```
/* Instance Variables */
ASFObjectFactory   ipronv_factory
. . .
/* assume that n_some_object is defined */
n_some_object   lnv_object

lnv_object = ipronv_factory.TRIGGER STATIC FUNCTION &
   CreateObject( "n_some_object" )
```

See Also

ASFObjectFactory

ASFService (abstract)

Ancestor Hierarchy

nonvisualobject

ASFObject

ASFKernel

ASFUtility

ASFService

Declared In

ap1kernl.pbl

Class Description

The ASFService class is the base class for service classes. A service class provides a single method, a service, to client classes. Descendents of the ASFService class can be used alone, but they are intended to be used with the ASFServiceManager and ASFNamedServiceManager classes.

Attributes

None.

Relations

None.

Class Invariant

External

None.

Internal

None.

Suggested Uses

Inherit from the ASFService class when there are many objects of a similar type (e.g., windows) that need a method (e.g., resize) once in a while. Rather than code the service as part of the win-

dow class, a service can provide the method and be immediately destroyed.

Method Categories

Misc. Methods

PerfomService()

Functions

PerfomService()

PUBLIC PerfomService (aany_requestor : ANY) : BOOLEAN

Description

This method is intended to be implemented by descendent classes. The method performs some service for the requesting object.

Parameters

aany_requestor	ANY - Reference to the object requesting the service.

Returns

BOOLEAN	TRUE if successful, FALSE otherwise.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

ASFStringAttribute

Ancestor Hierarchy

nonvisualobject

- **ASFObject**
  - **ASFKernel**
    - **ASFUtility**
      - **ASFAttribute**
        - **ASFStringAttribute**

Declared In

ap1kernl.pbl

Class Description

The ASFStringAttribute class holds an attribute name-value pair. Both the name and the value are strings. This class is used as a utility for managing many string attributes.

Attributes

Protected

ipros_value	STRING - The value of the attribute.

Relations

None.

Class Invariant

External

None.

Internal

None.

Suggested Uses

Whenever a named string is needed.

See Also

ASFAttribute
ASFMessageArguments
ASFMessageStringArguments

Method Categories

Get

GetAttributeValue()

Set

SetAttributeValue()

Functions

GetAttributeValue()

PUBLIC GetAttributeValue () : STRING

Description

This method gets the value of the attribute.

Parameters

None.

Returns

STRING	Attribute Value. In case of error, the empty string "" is returned.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

Example

```
ASFStringAttribute   lnv_attribute
STRING               ls_attribute value
STRING               ls_attribute_name
BOOLEAN              lb_return

lnv_attribute = CREATE ASFStringAttribute

lb_return = lnv_attribute.TRIGGER STATIC FUNCTION &
   SetAttributeName( "The Attribute Name" )

lb_return = lnv_attribute.TRIGGER STATIC FUNCTION &
   SetAttributeValue( "Some attribute value" )
. . .
ls_attribute_name = lnv_attribute.TRIGGER STATIC FUNCTION &
   GetAttributeName()

ls_attribute_value = lnv_attribute.TRIGGER STATIC FUNCTION &
```

```
GetAttributeValue()
```

See Also

SetAttributeValue()

SetAttributeValue()

PUBLIC SetAttributeValue (as_value : STRING) : BOOLEAN

Description

This method sets the attribute value.

Parameters

as_value	STRING - The string to which to set the attribute value.

Returns

BOOLEAN	TRUE if successful, FALSE otherwise.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

The string argument is NULL.

Example

```
ASFStringAttribute   lnv_attribute
STRING               ls_attribute value
STRING               ls_attribute_name
BOOLEAN              lb_return

lnv_attribute = CREATE ASFStringAttribute

lb_return = lnv_attribute.TRIGGER STATIC FUNCTION &
   SetAttributeName( "The Attribute Name" )

lb_return = lnv_attribute.TRIGGER STATIC FUNCTION &
   SetAttributeValue( "Some attribute value" )
. . .
ls_attribute_name = lnv_attribute.TRIGGER STATIC FUNCTION &
   GetAttributeName()

ls_attribute_value = lnv_attribute.TRIGGER STATIC FUNCTION &
   GetAttributeValue()
```

See Also

GetAttributeValue()

ASFSystemFunctions

Ancestor Hierarchy

nonvisualobject

- **ASFObject**
  - **ASFKernel**
    - **ASFUtility**
      - **ASFSystemFunctions**

Declared In

ap1kernl.pbl

Class Description

The ASFSystemFunctions class is a utility class with widely used functions that are not specific to any one class.

Attributes

None.

Relations

None.

Class Invariant

External

None.

Internal

None.

Method Categories

Misc.

IsValidObject()
StringToArray()
StringToBoolean()

Functions

StringToBoolean()

PUBLIC StringToBoolean (as_text : STRING, ab_default : BOOLEAN) : BOOLEAN

Description

Converts the text string, as_text, to its boolean equivalent. The following values for as_text return TRUE:

"true"

t

yes

y

on

The following values for as_text return FALSE

"false"

f

no

n

off

The string match is case insensitive. If as_text is not one of the above strings, then the default value ab_default is returned.

Parameters

as_text	STRING - Text to convert to BOOLEAN value.
ab_default	BOOLEAN - Default BOOLEAN value to return if the string as_text is not recognized.

Returns

BOOLEAN	Value of text conversion or default value.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

StringToArray()

PUBLIC StringToArray (as_text : STRING, as_delimiter : STRING, as_array[] : STRING) : INTEGER

Description

Parse the request string as_text and put the arguments into the string array as_array[]. The delimiter as_delimiter is used to properly parse the request string.

The arguments are passed back in the array as_array. The first argument (as_array[1]) contains the first element of the string. Also, the arguments of trimmed of white space in the beginning ending of the string.

Example:

as_text = 'Request Task:: DWC Parts :: Delete'

Returns:

as_array[1] = 'Request Task'

as_array[2] = 'DWC Parts'

as_array[3] = 'Delete'

Parameters

as_text	STRING - The string to be parsed.
as_delimiter	STRING - The delimiter separating string parts.
as_array[]	STRING - Array of string parts.

Returns

INTEGER	Number of elements in the string array

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

IsValidObject()

PUBLIC IsValidObject (aany_object : ANY) : BOOLEAN

Description

Tests object validity by using the PowerBuilder methods IsNull() and IsValid().

OBSOLETE METHOD: Use method IsValidObject() which is part of every AdaPT class.

Parameters

aany_object	ANY - The reference to the object to be tested for validity.

Returns

BOOLEAN	If IsNull() is TRUE or IsValid() is FALSE, then FALSE is returned. Otherwise, TRUE is returned.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

Example

```
/* assume anv_object is an argument */
BOOLEAN             lb_return
ASFSystemFunctions  lnv_sys_funcs

lb_return = FALSE

lnv_sys_funcs = CREATE ASFSystemFunctions

lb_return = lnv_sys_funcs.TRIGGER STATIC FUNCTION &
   IsValidObject( anv_object )

IF ( NOT IsValidObject( lb_return ) ) THEN
   RETURN lb_return
END IF
. . .
```

The Problem Domain Partition

ASFBusinessContext (abstract)

Ancestor Hierarchy

nonvisualobject

ASFObject

ASFDomain

ASFBusiness

ASFBusinessContext

Declared In

ap1pdp.pbl

Class Description

The ASFBusinessContext class is the base class for application business context classes. A business context provides the context in which business-related classes perform. The ASFBusinessContext class is intended to be the public interface between the User Interface and classes that perform business functions. The class also acts as a controller for business entities (objects) by managing the sequence of business events and routine events to the appropriate business entities. The ASFBusinessContext class communicates changes to business data to other parts of the applications, depending upon the application architecture.

Attributes

None.

Relations

Protected

ipronv_business_entity_manager	ASFBusinessEntityManager - Business entity (named) manager.

Class Invariant

External

None.

Internal

The instance variable ipronv_business_entity_manager must be valid. It is automatically created in the ASFBusinessContext constructor and destroyed in its destructor. Available method check for the validity of the variable ipronv_business_ entity _manager.

```
IF ( IsNull( ipronv_business_entity_manager )  ) THEN
   <some error code>
END IF

IF ( NOT IsValid( ipronv_business_entity_manager ) ) THEN
   <some error code>
END IF
```

Suggested Uses

The ASFBusinessContext class is used to manage one or more ASFBusinessEntity objects. This class provides a public interface to a specific business context, which can be equated to a business use case.

See Also

ASFBusinessEntity
ASFBusinessEntityManager

Method Categories

Events

Constructor()
Destructor()

Messaging

ParseTaskRequest()
ProcessTaskRequest()

Obsolete

RequestDataWindow()
ShareDataStore()

Registration

pronf_register_business_object()
pronf_unregister_business_object()

Functions

RequestDataWindow()

PUBLIC RequestDataWindow (as_dwc_name : STRING) : ASFDATAWINDOWCONTROL

Description

Request a reference the registered object of type ASFDataWindowControl with the registration name as_dwc_name. This method uses the business object manager (ipronv_business_object_manager) to get the requested DataWindow control from the appropriate business entity.

OBSOLETE METHOD (3/31/97): This method will be removed in a future version of AdAPT.

Parameters

as_dwc_name	STRING - Registration name of the requested DataWindow control.

Returns

ASFDATAWINDOW-CONTROL	Reference to the requested DataWindow control.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

NULL is returned if there is an error.

Example

```
/* Instance Variables */
PROTECTED:
ASFBusinessContext ipronv_business_context

/* Constants */
PROTECTED:
CONSTANT STRING ics_dwc_customer = "DWC Customer"
. . .
ASFDataWindowControl   luo_dwc
. . .
luo_dwc = ipronv_business_context.TRIGGER STATIC FUNCTION &
```

```
RequestDataWindow( ics_dwc_customer )
```

ProcessTaskRequest() (virtual)

PUBLIC ProcessTaskRequest (as_request : STRING) : BOOLEAN

Description

The purpose of this method is to Process the task request embodied in as_request.

Descendent classes must implement this method.

Parameters

as_request	STRING - The task request to be processed.

Returns

BOOLEAN	This method will always return FALSE.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

Example

Following is an example of ProcessTaskRequest() in a descendent of ASFBusinessContext. The following instance variables are also assumed:

PROTECTED:

STRING ipros_args[]

CONSTANT STRING icpros_delimiter = "::"

In this example, the request strings are in the following format:

<request type>::<request>::<parameter1>::. . .::<parameterN>

After the request string is parsed, then the first element of the argument array (as_args[1]) contains the request type, the second element (as_args[2]) holds the request or command, and subsequent elements contain any parameters passed. In the following example, the possible request types are "UserRequest" and "BusinessRequest".

```
/* ProcessTaskRequest() example */
BOOLEAN   lb_return

/* parse the request */
lb_return = THIS.TRIGGER STATIC FUNCTION &
   ParseTaskRequest( as_request, ipros_args[], icpros_delimiter )

IF ( NOT lb_return ) THEN
   RETURN lb_return
END IF

lb_return = FALSE
IF ( UpperBound( ipros_args ) < 1 ) THEN
   RETURN lb_return
END IF

CHOOSE CASE ipros_args[1]

   CASE "UserRequest"
      lb_return = THIS.TRIGGER STATIC FUNCTION &
         pronf_process_user_request()

   CASE "BusinessRequest"
      lb_return = THIS.TRIGGER STATIC FUNCTION &
         pronf_process_business_request()

   CASE ELSE
      lb_return = FALSE

END CASE

RETURN lb_return
```

In this example of ProcessTaskRequest(), the request is parsed and the string arguments are stored in the instance variable ipros_args[]. A CHOOSE CASE statement is used to handle the request, in this case using the functions defined in this example descendent class (pronf_process_user_request() and pronf_process_business_request()).

See Also

ParseTaskRequest()

ParseTaskRequest()

PUBLIC ParseTaskRequest (as_request : STRING, as_args[] : REF STRING, as_delimiter : STRING) : BOOLEAN

Description

This method parses the request string as_request and put the arguments into the string array as_args. The delimiter as_delimiter is used to properly parse the request string.

The arguments are passed back in the array as_args. The first argument (as_args[1]) contains the first element of the string. Also, the arguments are trimmed of white space at the beginning and ending of the string. For example, if:

as_request = "Request Task:: DWC Parts :: Delete"

then the following is return returned in as_args[]:

as_args[1] = "Request Task"

as_args[2] = "DWC Parts"

as_args[3] = "Delete"

Parameters

as_request	STRING - The request string to be parsed.
as_args[]	REF STRING - The string array in which the separated arguments are passed back.
as_delimiter	STRING - The delimiter used in the request string to separate arguments.

Returns

BOOLEAN	TRUE if successful, FALSE otherwise.

Precondition

The parameter as_request is not null and is valid.

Postcondition

See class invariant.

Exception

None.

Example

```
BOOLEAN   lb_return
STRING    ls_request
STRING    ls_args[]

CONSTANT STRING    lcs_delimiter = "::"
. . .
ls_request = "Request Task::  DWC Parts  :: Delete"
lb_return = THIS.TRIGGER STATIC FUNCTION &
   ParseTaskRequest( ls_request, ls_args[], lcs_delimiter )

Returns

ls_args[1] = "Request Task"
ls_args[2] = "DWC Parts"
ls_args[3] = "Delete"
```

See Also

ProcessTaskRequest()

ShareDataStore()

PUBLIC ShareDataStore (as_ds_name : STRING, auo_dwc : REF ASFDATAWINDOWCONTROL) : BOOLEAN

Description

Share the data of the DataStore referenced by as_ds_name with the DataWindow control auo_dwc. The DataStore is the primary object in the share relationship and the DataWindow control is secondary. The business object manager(ipronv_business_object_manager) is used to find the business entity with the appropriate DataStore with which to share data.

OBSOLETE METHOD (3/31/97): This method will be removed in a future version of AdAPT.

Parameters

as_ds_name	STRING - Name of the DataStore to share data.
auo_dwc	REF ASFDATAWINDOWCONTROL - The DataWindow control to be the secondary control in the share relationship.

Returns

BOOLEAN	TRUE if successful, FALSE otherwise.

Precondition

See class invariant.

Postcondition

See class invariant.

Exception

None.

Example

```
/* Instance Variables */
PROTECTED:
ASFBusinessContext   ipronv_business_context
ASFDataWindowControl iprouo_dwc

/* Constants */
PROTECTED:
CONSTANT STRING ics_ds_customer = "DS Customer"
. . .
BOOLEAN   lb_return
```

```
lb_return = ipronv_business_context.TRIGGER STATIC FUNCTION &
   ShareDataStore( ics_ds_customer, iprouo_dwc )
```

See Also

DataWindowControlAdapter()

DataStoreAdapter()

pronf_register_business_object()

PROTECTED pronf_register_business_object (as_business_object_type : STRING,
as_business_object_name : STRING) : STRING

Description

This method registers a business object with the business entity manager. The intent is to register an object of type ASFBusinessEntity or ASFBusinessContext, or one of their descendants. The object descendent from ASFBusinessContext (THIS) is the requesting object. The registration is set with the sole-use-only flag set to TRUE and the persistence flag set to FALSE.

Parameters

as_business_object_type	STRING - The type of object to be registered.
as_business_object_name	STRING - The name of the object to be registered.

Returns

STRING	Registration ID of the new registration. An empty string ("") indicates an error.

Precondition

Any type of object may be registered because it is not possible to check the class types. If the object type is not a descendent of ASFBusiness, then there will be run-time errors dealing with mismatched references.

Postcondition

See class invariant.

The object of type as_business_object_type is not immediately created. Instead, it is registered and not created until the first time the registration is accessed using the RequestComponent() method.

The registration sole-use-only flag is set to TRUE and the persistence flag is set to FALSE.

Exception

None.

Example

The following example assumes the existence of class n_bo_customer:

```
/* Instance Variables */
```

```
PROTECTED:
STRING    ipros_business_object_type
STRING    ipros_business_object_name
. . .
STRING   ls_return

ipros_business_object_type = "n_bo_customer"
ipros_business_object_name = "customer"
ls_return = THIS.TRIGGER STATIC FUNCTION &
   pronf_register_business_object( &
ipros_business_object_type, ipros_business_object_name )
```

See Also

pronf_unregister_business_object()